

Amendment to the Specification:

Please replace paragraph [0001] with the following amended paragraph:

[0001] This application claims the benefit of U.S. Provisional Application Nos. 60/408,735 and 60/409,311, both filed September 6, 2002, which are incorporated herein by this reference. This application is related to U.S. Application Serial Nos. 10/655,951 and 10/655,961, both filed September 4, 2003, which are incorporated herein by this reference.

Please add the following new paragraphs after paragraph [0021] and before paragraph [0022]:

[0021.1] FIGURE 11 illustrates a snapshot tree descended from a base volume.

Please replace paragraph [0036] with the following amended paragraph:

[0036] Additional details regarding the scalable cluster data handling system and its nodes are provided in co-pending U.S. Patent Application Serial No. 09/633,088, entitled "Data Storage System," (~~Attorney Docket No. M-8494~~) filed on August 4, 2000; U.S. Patent Application Serial No. 09/883,681, entitled "Node Controller For A Data Storage System," (~~Attorney Docket No. M-8496~~) filed on June 18, 2001; and U.S. Patent Application No. 10/655,951, entitled "Time And Space Efficient Technique for Creating Virtual Volume Copies," (~~Attorney Docket No. 3PD-M-8497-US~~), filed concurrently. These applications are assigned to the same Assignee as the present application and are hereby incorporated by reference in their entireties.

Please replace paragraph [0096] with the following amended paragraph:

[0096] Additional details regarding the tree-like data structure and its advantages are provided in co-pending U.S. Provisional Patent Application No. 10/655,961, entitled "Read-Write Snapshots," (~~Attorney Docket No. 3PD-P100~~) filed concurrently. Such application is assigned to the same Assignee as the present application and is hereby incorporated by reference in its entirety.

Please add the following new paragraphs after paragraph [0121] and before paragraph [0122]:

[0121.1] Fig. 11 illustrates a snapshot tree 2000 for a virtual volume 1026. One or more snapshot trees may be generated or provided for each virtual volume 1026 that is maintained in a data storage system. As depicted, snapshot tree 2000 includes a base virtual volume 2200 and a series of snapshot volumes 2104, 2106, 2204, 2210, 2212, 2206, 2304, 2308, 2310, and 2306.

[0121.2] Base virtual volume 2200 can be written into and read from by a user or host device 18. The base virtual volume 2200 may be the most current version of the respective virtual volume 1026, and most reads and writes of data are performed on the base virtual volume. From another perspective, base virtual volume 2200 comprises data initially stored at a point in time, such as time X (or time 0), and any data that has been subsequently written by a host device or user after time X. Base virtual volume 2200 may serve as a “root” for the snapshot tree 2000.

[0121.3] Each snapshot volume maintains data and tables for an associated snapshot of the base virtual volume. As such, for snapshot tree 2000, the snapshot volumes may be considered to “descend” from a base virtual volume 2200. Any of the snapshot volumes can be accessed to obtain data that was written at a prior time. A snapshot volume can be either a read-only (R/O) snapshot volume (or ROSS) or a read/write (R/W) snapshot volume (or RWSS). A ROSS presents a constant view of the data in a virtual volume at a specific time. After creation of a ROSS, data can be read from but not written into the ROSS. A RWSS descends from a ROSS (e.g., a parent snapshot volume) and may serve to hold modifications to the parent ROSS. A RWSS can be read and written like a base virtual volume. As such, a RWSS can be viewed as a writable/modifiable version of its parent ROSS. As shown, snapshot volumes 2106, 2204, 2210, 2212, 2206, 2308, and 2310 are ROSSes, and snapshot volumes 2104, 2304, and 2306 are RWSSes. Each of the RWSSes may have one or more descending ROSSes. As can be seen, for example, a RWSS 2306 can descend from a ROSS 2308 of another RWSS 2304.

[0121.4] The snapshot volumes may be grouped in branches. A branch is made up of a read-write volume (either base virtual volume or RWSS) as its base and one or more read-only snapshot volumes maintained in a time-ordered link attached to the read-write volume. Thus, referring to Fig. 11 for example, a branch can be the base virtual volume 2200 and a sequence of read-only snapshot volumes, such as the ROSSes 2204, 2210, 2212, and 2206. A branch may also be a read/write snapshot volume, such as a RWSS 2304 and one or more read-only snapshot volumes, such as the ROSSes 2308 and 2310. A new branch can be created by adding a read-write snapshot volume to a read-only snapshot volume, after which read-only snapshot volumes can be added to grow the branch. For any given branch, the snapshot volumes extend from oldest to most recent. For example, in the branch comprising base volume 2200 and snapshot volumes 2204, 2210, 2212, and 2206, snapshot volume 2206 is the oldest (created earliest in time) while snapshot 2204 is the most recent (created last in time).

[0121.5] A snapshot volume may be created or started by execution of a command from the volume manager 102, a node 12, or a host device 18. For example, at one point in time, the volume manager 102 may execute a command that causes the creation of a first snapshot volume (e.g., snapshot volume 2206). At subsequent points in time, the volume manager 102 may execute other commands which can similarly result in creation of additional snapshot volumes (e.g., snapshot volumes 2212, 2210, and 2204). Thus, for example, a second snapshot volume (e.g., snapshot volume 2212) stores data that has been more recently changed or modified than data in the first snapshot volume.

[0121.6] If return to a particular state of memory is desired, a snapshot volume corresponding to the snapshot for the particular state is accessed. In this way, copy-on-write operations can be reversed.

[0121.7] In one embodiment, data of a virtual volume may be read in a manner opposite to a write, for example, by accessing the data from the base virtual volume 2200 and one or more snapshot volumes, as desired. A data block from the base virtual volume 2200 may be accessed by simply reading the physical storage designated by the base virtual volume mappings (e.g., from a virtual volume region table 104). A data block from a snapshot volume may be accessed by reading through the level mapping tables (i.e., from the Level 1 table to the last level table, such as, for example, Level 3 table). If the entries in the tables of the snapshot volume associated with the data block are not zero, then a pointer for that element exists and can be used to read the stored data from the snapshot volume.

[0121.8] Each of the volumes in the snapshot tree from one that is under consideration up to the base volume may be analyzed in turn to see whether the data block was modified during the respective snapshots. The snapshot volumes are read until a pointer value that is not zero is available or the base virtual volume is reached. That is, for each snapshot volume, the data storage system passes level by level through the storage structures until a pointer is available, and reads the data designated by the first available pointer. If the data block was not found in any of the snapshot volumes, then the system looks in the base virtual volume.

[0121.9] In another embodiment, a snapshot read operation is performed by first accessing the data structures of the most recent snapshot volume before the data structures of the base virtual volume so that the latest written data is accessed. In a read of the snapshot volumes, the system

searches the various Level 1, 2, and so on tables, and if a pointer entry is found in the snapshot volume, the entry is returned as the result of the read operation. A pointer in the final level table (e.g., Level 3 tables 670a, 670b, or 670c) points to a block in physical storage. If no pointer entry is found in the snapshot volumes, the system returns to the base virtual volume.

[0121.10] In some embodiments, pointers may be set to skip over one or more snapshot volumes of the snapshot tree. For example, if a desired data block is found in the fourth snapshot volume along a branch, then a pointer may be set in the first snapshot volume so that a subsequent search for the data block in the first snapshot volume will automatically skip to the fourth snapshot volume. This saves time and improves performance by avoiding the second and third snapshot volumes in subsequent searches for that data block.

[0121.11] In one embodiment, data and structures of the base and snapshot volumes of the snapshot tree 2000 may be exported or transferred between nodes 12 of the data storage system.

[0121.12] Additional details regarding the tree-like data structure and its advantages are provided in co-pending U.S. Patent Application No. 10/655,961, entitled "Read-Write Snapshots," filed concurrently. Such application is assigned to the same Assignee as the present application and is hereby incorporated by reference in its entirety.